2004

Robust and Distributed Computation of Aggregates in Wireless Sensor Networks

Jen-Yeu Chen

Gopal Pandurangan

Dongyan Xu
Purdue University, dxu@cs.purdue.edu

Report Number:
04-032

ROBUST AND DISTRIBUTED COMPUTATION OF AGGREGATES IN WIRELESS SENSOR NETWORKS

Jen-Yeu Chen
Gopal Pandurangan
Dongyan Xu

Department of Computer Sciences
Purdue University
West Lafayette, IN 47907

CSD TR #04-032
December 2004

Robust and Distributed Computation of Aggregates in Wireless Sensor Networks

Jen-Yeu Chen, Gopal Pandurangan, Dongyan Xu

Abstract

A wireless sensor network consists of a large number of small, resource-constrained devices and usually operates in hostile environments that are prone to link and node failures. Computing aggregates such as average, minimum, maximum and sum is fundamental to various primitive functions of a sensor network like system monitoring, data querying, and collaborative information processing. In this paper we present and analyze a suite of randomized distributed algorithms to efficiently and robustly compute aggregates. Our *Distributed Random Grouping (DRG)* algorithm is simple and natural and uses probabilistic grouping to progressively converge to the aggregate value. DRG is local and randomized and is naturally robust against dynamic topology changes from link/node failures. Although our algorithm is natural and simple, it is nontrivial to show that it converges to the correct aggregate value and to bound the time needed for convergence. Our analysis uses the eigen-structure of the underlying graph in a novel way to show convergence and to bound the running time of our algorithms. We also present simulation results of our algorithm and compare its performance to various other known distributed algorithms. Simulations show that DRG needs much less transmissions than other distributed localized schemes.

Index Terms

Probabilistic algorithms, Randomized algorithms, Distributed algorithms, Sensor networks, Fault tolerance, Graph theory, Aggregate, Data query, Stochastic processes.

I. Introduction

Sensor nodes are usually deployed in hostile environments. As a result, nodes and communication links are prone to failure. This makes centralized algorithms undesirable in sensor networks with resource limited sensor nodes[2], [7], [27]. In contrast, localized distributed algorithms are simple, scalable, and robust to network topology changes as nodes only communicate with their neighbors[2], [4].

For cooperative processing in a sensor network, the information of interest is not the data at an individual sensor node, but the aggregate statistics (*aggregates*) amid a group of sensor nodes[19], [12], [18]. Possible applications using aggregates are the average temperature, the average gas concentration of a hazardous gas in an area, the average or minimum remaining battery life of sensor nodes, the count of some endangered animal in an area,

The authors are in the alphabetical order of last names and with Department of Computer Science, Purdue University, USA. E-mail: jenyeu@ieee.org.; {gopal,dxu}@cs.purdue.edu

and the maximal noise level in a group of acoustic sensors, to name a few. The operations for computing basic aggregates like average, max/min, sum, and count could be further adapted to more sophisticated data query or information processing operations[8], [22], [20], [21]. For instance, the function $f(\mathbf{v}) = \sum c_i f_i(v_i)$ is the *sum* aggregate of values $c_i f_i(v_i)$ which are pre-processed from v_i on all nodes.

In this paper, we present and analyze a simple, distributed, localized, and randomized algorithm called as *Distributed Random Grouping (DRG)* to compute *aggregate* information in wireless sensor networks. DRG is more efficient than another randomized distributed algorithm, Uniform Gossip[7], because DRG takes advantage of the broadcast nature of wireless transmissions. All nodes within the radio coverage can hear and receive a wireless transmission. Although broadcast-based Flooding[7] also exploits the broadcast nature of wireless transmission, on some network topologies like grid, it is possible that the aggregate computing will not correctly converge. We suggest a modified broadcast-based Flooding, Flooding-m, to mitigate this pitfall and compare it with DRG by simulations.

Deterministic tree-based in-network approaches have been successfully developed to compute aggregates[19], [20], [21]. In [7], [11], it is shown that tree based algorithms face challenges in efficiently maintaining resilience to topology changes. The authors of [19] have addressed the importance and advantage of in-network aggregation. They build an optimal aggregation tree to efficiently computed the aggregates. Their *centralized* approaches are heuristic since building an optimal aggregation tree in a network is the Minimum Steiner Tree problem, known to be NP-Hard[19]. Although a *distributed* heuristic tree approach [54] could save the cost of *coordination* at the tree construction stage, the aggregation tree will need to be reconstructed whenever the topology changes, before aggregate computation can resume or re-start. The more often the topology changes, the more overhead that will be incurred by the tree reconstruction. On the other hand, distributed localized algorithms such as our proposed DRG, Uniform Gossip [7], and Flooding [7] are *free* from the global data structure maintenance. Aggregate computation can continue without being interrupted by topology changes. In contrast to tree-based approaches that obtain the aggregates at a single (or a few) sink node, these algorithms converge with *all* nodes knowing the aggregate computation results. In this way, the computed results become robust to node failures, especially the failure of sink node or near-sink nodes. In tree based approaches the single failure of sink node will cause loss of all computed aggregates. Also, it is convenient to retrieve the aggregate results, since all nodes have them. In mobile-agent-based sensor networks[28], [29], [30], this can be especially helpful when the mobile agents need to stroll about the hostile environment to collect aggregates.

Although our algorithm is natural and simple, it is nontrivial to show that it converges to the correct aggregate value and to bound the time needed for convergence. Our analysis uses the eigen-structure of the underlying graph in a novel way to show convergence and to bound the running time of our algorithms. We use the algebraic connectivity of the underlying graph, the second smallest eigenvalue of the Laplacian matrix of the graph, to bound the running time and total number of transmissions. The performance analysis of the average aggregate computation by *DRG Ave* algorithm is our main analysis result. We also extend it to the analysis of global *maximum* or *minimum* computation. We also provide analytical bounds for convergence assuming wireless link failures. Other aggregates

such as sum and count can be computed by running an adapted version of DRG Ave.

II. Related Work

The Uniform Gossip algorithm, Push-Sum, [7] is a distributed algorithm to compute the average on sensor and P2P networks. Under the assumption of a *complete graph*, their analysis shows that with high probability the values at all nodes converges exponentially fast to the true (global) average.[1] The authors of [7] point out that the point-to-point Uniform Gossip protocol is not suitable for wireless sensor or P2P networks. They propose an alternative distributed broadcast-based algorithm, Flooding, and analyze its convergence by using the mixing time of the random walk on the underlying graph. Their analysis assumes that the underlying graph is ergodic and reversible (and hence their algorithms may not converge on many natural topologies such as Grid — see Fig.6 for a simple example). However, the algorithm runs very fast (logarithmic in the size) in certain graphs, e.g., on an expander, which is however, not a suitable graph to model sensor networks. Also, their analysis of Uniform Gossip and Flooding did not consider possible collisions among wireless transmissions.

The authors of [31] discuss distributed algorithms for computations in ad-hoc networks. They have a deterministic and distributed *uniform diffusion* algorithm for computing the average. They set up the convergence condition for their uniform diffusion algorithm. However, they do not give a bound on running time. They also find the optimal diffusion parameter for each node. However, the execution of their algorithm needs global information such as maximum degree or the eigenvalue of a topology matrix. Our DRG algorithms are purely local and do not need any global information, although some global information is used (only) in our analysis.

Randomized gossiping in [56] can be used to compute the aggregates in arbitrary graph since at the end of gossiping, all the nodes will know all others' initial values. Every node can post-process all the information it received to get the aggregates. The bound of running time is $O(n \log^3 n)$ in arbitrary directed graphs. However, this approach is not suitable for resource-constrained sensor networks, since the number of transmission messages grows *exponentially*.

Finally, we mention that there have been some works on flocking theory (e.g., [57]) in control systems literature; however, the assumptions, details, and methodologies are very different from the problem we address here.

III. Overview

A sensor network is abstracted as a connected undirected graph $G(\mathcal{V}, \mathcal{E})$ with all the sensor nodes as the set of vertices $\mathcal{V}$ and all the bi-directional wireless communication links as the set of edges $\mathcal{E}$. This underlying graph can be arbitrary depending on the deployment of sensor nodes.

Let each sensor node i be associated with an initial observation or measurement value denoted as $v_i^{(0)}$ ($v_i^{(0)} \in \mathbb{R}$). The assigned values over all vertices is a vector $\mathbf{v}^{(0)}$. Let $v_i^{(k)}$ represent the value of node i after running our algorithms for k rounds. For simplicity of notation, we omit the superscript when the specific round number k doesn't matter.

[1]The unit of running time is the synchronous round among all the nodes.

The goal is to compute (aggregate) functions such as average, sum, max, min etc. on the vector of values $\mathbf{v}^{(0)}$. In this paper, we present and analyze simple and efficient, robust, local, distributed algorithms for the computation of these aggregates.

The main idea in our algorithm, *random grouping* is as follows. In each "round" of the algorithm, every node independently becomes a group leader with probability p_g and then invites its neighbors to join the group. Then all members in a group update their values with the locally derived *aggregate* (average, maximum, minimum, etc) of the group. Through this randomized process, we show that all values will progressively converge to the correct aggregate value (the average, maximum, minimum, etc.). Our algorithm is distributed, randomized, and only uses local communication. Each node makes decisions independently while all the nodes in the network progressively move toward a consensus.

To measure the performance, we assume that nodes run DRG in synchronous time slots, i.e., rounds, so that we can quantify the running time. The synchronization among sensor nodes can be achieved by applying the method in [14], for example. However, we note that synchronization is not crucial to our approach and our algorithms will still work in an asynchronous setting, although the analysis will be somewhat more involved.

Our main technical result gives an upper bound on the expected number of rounds needed for all nodes running DRG Ave to converge to the *global average*. The upper bound is

$$O(\frac{1}{\gamma}\log(\frac{\phi_0}{\varepsilon^2})),$$

where the parameter γ directly relates to the properties of the graph, and the grouping probability used by our randomized algorithm; and ε is the desired accuracy (all nodes' values need to be within ε from the global average). The parameter ϕ_0 represents the grand variance of the initial value distribution.

The upper bound on the expected number of rounds for computing the global maximum or minimum is

$$O(\frac{1}{\gamma}\log(\frac{(1-\rho)n}{\rho})),$$

where ρ is the accuracy requirement for Max/Min problem (ρ is the ratio of nodes which do *not* have the global Max/Min value to all nodes in the network). A bound for the expected number of necessary transmissions can be derived by using the result of the bound on the expected running time.

The rest of this paper is organized as follows. In section IV, we detail our distributed random grouping algorithm. In section V we analyze the performance of the algorithm while computing various aggregates such as average, max, and min. In section VI, we discuss practical issues in implementing the algorithm. The extensive simulation results of our algorithm and the comparison to other distributed approaches of aggregates computation in sensor network are presented in section VII. Finally, we conclude in section VIII.

IV. ALGORITHMS

Fig. 1 is a high-level description of *DRG Ave* for global average computation. The description in Fig. 1 does not assume the synchronization among nodes whereas for analysis we assume nodes work in synchronous rounds. A round contains all the steps in Fig. 1.

Alg: DRG Ave: Distributed Random Grouping for Average

1.1 Each node in ***idle mode*** independently originates to form a group and become the group leader with probability p_g.

1.2 A node i which decides to be the group leader enters the ***leader mode*** and broadcasts a group call message, $GCM \equiv (group_{id} = i)$, to all its neighbors.

1.3 The group leader i waits for responses message, $JACK$ from its neighbors.

2.1 A neighboring node j, at the ***idle mode*** that successfully received the GCM, responds to the group leader i an joining acknowledgement, $JACK \equiv (group_{id} = i, v_j, join(j) = 1)$, with its value v_j.

2.2 The node j enters ***member mode*** and waits for the group assignment message GAM from its leader.

3.1 The group leader, node i, gathers the received $JACKs$ from its neighbors to compute the number of group members, $J = \sum_{j \in g_i} join(j)$, and the average value of the group, $Ave(i) = \frac{\sum_{k \in g_i} v_k}{J}$.

3.2 The group leader, node i, broadcasts the group assignment message $GAM \equiv (group_{id} = i, Ave(i))$ to its group members and then returns to ***idle mode***.

3.3 A neighboring node j, at ***member mode*** of the group i which receives GAM, updates its value $v_j = Ave(i)$ and then returns to ***idle mode***.

Fig. 1. DRG Ave algorithm

Each sensor node can work in three different modes, namely, idle mode, leader mode, and member mode. A node in idle mode becomes a group leader with probability p_g, or remains idle with probability $(1 - p_g)$. Choosing a proper p_g will be discussed in Section V.

A group leader announces the Group Call Message (GCM) by a wireless broadcast transmission. The Group Call Message includes the leader's identification as the group's identification. An idle neighboring node who successfully receives a GCM then responds to the leader with a Joining Acknowledgement (JACK) and becomes a member of the group. The JACK contains the sender's value for computing aggregates. After sending JACK, a node enters member mode and will not response to any other GCMs until it returns to idle mode again. A member node waits for the local aggregate from the leader to update its value. The leader gathers the group members' values from JACKs, computes the local aggregate (average of its group) and then broadcasts it in the Group Assignment Message (GAM) by a wireless transmission. Member nodes then update their values by the assigned value in the received GAM. Member nodes can tell if the GAM is their desired one by the group identification in GAM.

The *DRG Max/Min* algorithms to compute the maximum or minimum value of the network is only a slight modification of the DRG Ave algorithm. Instead of broadcasting the local average of the group, in the step 3, the group leader broadcasts the local maximum or minimum of the group.

V. Analysis

In this section we analyze the DRG algorithms by two performance measurement metrics: expected running time and expected total number of transmissions. The number of total transmissions is a measurement of energy cost of the algorithm. The running time will be measured in the unit of a "round" which contains the three main steps in Fig. 1.

Our analysis builds on the technique of [6] which analyzes a problem of dynamic load balancing by random matchings. In the load balancing problem, they deal with discrete values (integer), but we deal with continuous values (real) which makes our analysis different. Our algorithm uses random groupings instead of random matchings. This has two advantages. The first we show that the convergence is faster and hence faster running time and more importantly, it is well-suited to the ad hoc wireless network setting because it is able to exploit the broadcast nature of wireless communication. Our random grouping algorithm is adapted to the ad hoc wireless setting and is mindful of problems such as collisions (due to interference of messages between nearby nodes).

The main result of this section is the following theorem.

***Theorem** 1:* Given a connected undirected graph $G(\mathcal{V}, \mathcal{E})$, $|\mathcal{V}| = n$ and an arbitrary initial value distribution $\mathbf{v}^{(0)}$ with the initial potential ϕ_0, then with high probability, the average problem can be solved by the DRG Ave algorithm with a $\varepsilon > 0$ accuracy, i.e., $|v_i - \bar{v}| \leq \varepsilon$, $\forall i$ in

$$O(\frac{d \log(\frac{\phi_0}{\varepsilon^2})}{p_g p_s (1+\alpha) a(G)})$$

rounds, where $a(G)$ is the algebraic connectivity (second smallest eigenvalue of the Laplacian Matrix of graph G[39], [40]); $d = \max(d_i) + 1 \approx \max(d_i)$ (the maximum degree); p_g is the grouping probability; and p_s is the probability of no collision of a group leader's group call message, GCM.

The constant $\alpha > 1$. The $\phi_0 = O(n)$ when all $v_i^{(0)}$ are bounded ($v_i^{(0)} < c$). Table I shows the algebraic connectivity $a(G)$ and $d/a(G)$ on several typical graphs.

The p_s is related to p_g and the graph's topology. Given a graph, an increasing p_g results in a decreasing p_s, and vice versa. However, there does exist a maximum value of $\mathcal{P} = p_g \cdot p_s$ so that we could have the best performance of DRG by a wise choice of p_g. We will discuss later about how to appropriately choose p_g to maximize $p_g p_s$ after proving the theorem.

The proof and the discussions of Theorem 1 are presented in the following paragraphs. To analyze our algorithm we need the concept of a *potential* function as defined below.

***Definition** 2:* Consider an undirected connected graph $G(\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| = n$ nodes. Given a value distribution $\mathbf{v} = (v_1, ..., v_n)^T$, v_i is the value on node i, the potential of the graph ϕ is defined as

$$\phi = ||\mathbf{v} - \bar{v}\mathbf{u}||_2^2 = \sum_{i \in \mathcal{V}} (v_i - \bar{v})^2 = (\sum_{i \in \mathcal{V}} v_i^2) - n\bar{v}^2$$

where $\bar{v}$ is the mean (global average) value over the network.

ϕ is a measurement of the grand variance of the value distribution. Note that $\phi = 0$ if and only if $\mathbf{v} = \bar{v}\mathbf{u}$, where $\mathbf{u} = (1, 1, \ldots, 1)^T$ is the unit vector. We will use the notation ϕ_k to denote the potential in round k and use ϕ in

TABLE I

THE ALGEBRAIC CONNECTIVITY $a(G)$ AND $d/a(G)$,[6]

Graph	$a(G)$	$d/a(G)$
Clique	n	$O(1)$
d-regular expander	$\Theta(d)$	$O(1)$
grid	$\Theta(\frac{1}{n})$	$O(n)$
linear array	$\Theta(\frac{1}{n^2})$	$O(n^2)$

general when specific round number doesn't matter.

Let the potential decrement from a group g_i led by node i after one round of the algorithm be $\delta\phi|_{g_i} \equiv \delta\varphi_i$,

$$\delta\varphi_i = \sum_{j\in g_i} v_j^2 - \frac{(\sum_{j\in g_i} v_j)^2}{J} = \frac{1}{J}\sum_{j,k\in g_i}(v_j - v_k)^2,$$

where $J = |g_i|$ is the number of members joining group i (including the leader i). The property $\delta\varphi_i \geq 0$ indicates that the value distribution $\mathbf{v}$ will eventually converge to the average vector $\bar{v}\mathbf{u}$ by invoking our algorithm repeatedly.

For analysis, we assume that every node independently but simultaneously decides whether to be a group leader at the beginning of a round. Those who decided to be leaders then send out their GCMs at the same time.

It is possible that a collision[2] happens between two GCMs so that some nodes within the overlap area of the two GCMs will not respond and join any groups. In the analysis below, we only consider *complete* groups which include all neighbors of the group leaders. This is a *worst* situation analysis which gives an upper bound. A tighter bound on Poisson random geometric graphs[50] by including both *partial* and *complete* groups is presented in subsection E of this section.

A. *Proof of Theorem 1*

The main thrust of the proof is to suitably bound the expected rate of decrement of the potential function ϕ.

We need a few definitions for the proof. We define the set $\tilde{\mathcal{N}}_G(i)$, including all members of a *complete group*, as $\tilde{N}_G(i) = \mathcal{N}_G(i) \cup \{i\}$ where the $\mathcal{N}_G(i) = \{j|(i,j) \in \mathcal{E}(G)\}$ is the set of neighboring nodes of leader i. Since we consider *complete groups* only, the set of nodes within a group $g_i = \tilde{\mathcal{N}}_G(i)$ is $J = |g_i| = d_i + 1$, where d_i is the degree of leader i. Let $C_i = G(\tilde{\mathcal{N}}_G(i)) = K_{d_i+1}$, be the $|\tilde{\mathcal{N}}(i)|$-clique on the set of nodes of $\tilde{\mathcal{N}}_G(i)$.

Define an auxiliary graph $H = \bigcup_{i\in\mathcal{V}(G)} C_i$ and the set of all auxiliary edges $\overline{\mathcal{E}} = \mathcal{E}(H) - \mathcal{E}(G)$. The Figure 2 shows a connected graph G, the groups led by each node of G as well as their associated cliques, and the auxiliary graph H. A real edge (x, y) of solid line in these graphs indicates that two end nodes, x and y can communicate

[2]It is also possible that a perfect MAC layer protocol can avoid collisions amid GCMs so that a node successfully received several GCMs can randomly choose one group to join. Since there is no standard MAC or PHY layer protocols in sensor networks, to analyze our algorithm in a general way, only complete groups are considered.

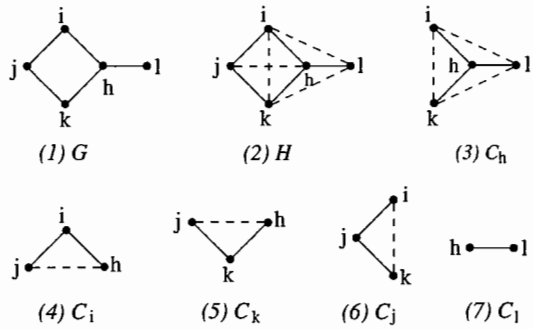


Fig. 2. graph G, the group Cliques of each node and the auxiliary graph H

with each other by the wireless link. The auxiliary edges are shown in dashed lines. These auxiliary edges are not real wireless links in the sensor network but will be helpful in the following analysis.

***Lemma** 3:* The convergence ratio

$$E[\frac{\delta\phi}{\phi}] \quad \geq \quad (1+\alpha)a(G)\frac{p_g p_s}{d}$$

where $\alpha \geq 1$ is a constant.

Proof: Let $x_i = (v_i - \bar{v})$, $\mathbf{x} = (x_1, ...x_n)^T$, $\phi = \mathbf{x}^T\mathbf{x}$, and Laplacian Matrix $\mathcal{L} = \mathcal{D} - \mathcal{A}$. The $\mathcal{D}$ is the diagonal matrix with $\mathcal{D}(v,v) = d_v$, the degree of node v; and the $\mathcal{A}$ is the adjacent matrix of the graph. $\mathcal{L}_G$ and $\mathcal{L}_H$ are the Laplacian Matrices of graph G and H respectively.

Let $\Delta_{jk} = (v_j - v_k)^2 = (x_j - x_k)^2$; p_s be the probability for a node to announce the GCM without collision, (e.g. in Poisson random geometric graph[50] $p_s = e^{-p_g \cdot \lambda \cdot 4\pi r^2}$); and $d = \max(d_i) + 1$, where d_i is the degree of node i. The expected decrement of the potential in the whole network is

$$\begin{aligned}
E[\delta\phi] &= E[\sum_{i\in\mathcal{V}} \delta\varphi_i] = p_g p_s \sum_{i\in\mathcal{V}} \delta\varphi_i \\
&= p_g p_s \sum_{i\in\mathcal{V}} \frac{1}{d_i+1} \sum_{(j,k)\in\mathcal{E}(C_i)} \Delta_{jk} \\
&\geq p_g p_s \frac{1}{d} \sum_{i\in V} \sum_{(j,k)\in\mathcal{E}(C_i)} \Delta_{jk} \qquad (1) \\
&= p_g p_s \frac{1}{d} \sum_{i\in\mathcal{V}} \sum_{(j,k)\in\mathcal{E}(C_i)} (x_j - x_k)^2 \\
&\geq p_g p_s \frac{1}{d} (\sum_{(j,k)\in\mathcal{E}(G)} 2(x_j - x_k)^2 + \sum_{(j,k)\in\overline{\mathcal{E}}} (x_j - x_k)^2)) \qquad (2) \\
&= p_g p_s \frac{1}{d} (\sum_{(j,k)\in\mathcal{E}(G)} (x_j - x_k)^2 + \sum_{(j,k)\in\mathcal{E}(H)} (x_j - x_k)^2) \\
&= p_g p_s \frac{1}{d} (\mathbf{x}^T\mathcal{L}_G\mathbf{x} + \mathbf{x}^T\mathcal{L}_H\mathbf{x}).
\end{aligned}$$

The equation (2) follows from the fact that for each edge $(i,j) \in \mathcal{E}$, Δ_{ij} appears at least twice in the sum $E[\delta\phi]$. Also each auxiliary edge $(j,k) \in \overline{\mathcal{E}}$ contributes at least once.

$$\begin{aligned} E[\frac{\delta\phi}{\phi}] &\geq p_g p_s \frac{1}{d}(\frac{\mathbf{x}^T\mathcal{L}_G\mathbf{x} + \mathbf{x}^T\mathcal{L}_H\mathbf{x}}{\mathbf{x}^T\mathbf{x}}) \\ &\geq p_g p_s \frac{1}{d}(\min_{\mathbf{x}}(\frac{\mathbf{x}^T\mathcal{L}_G\mathbf{x}}{\mathbf{x}^T\mathbf{x}}|\mathbf{x}\perp\mathbf{u}, \mathbf{x}\neq 0) + \min_{\mathbf{x}}(\frac{\mathbf{x}^T\mathcal{L}_H\mathbf{x}}{\mathbf{x}^T\mathbf{x}}|\mathbf{x}\perp\mathbf{u}, \mathbf{x}\neq 0)) \\ &= p_g p_s \frac{1}{d}(a(G) + a(H)) \;=\; (1+\alpha)a(G)\frac{p_g p_s}{d} \end{aligned}$$

In the above, we exploit the Courant-Fischer Minimax Theorem[40]:

$$a(G) = \lambda_2 = \min_{\mathbf{x}}(\frac{\mathbf{x}^T\mathcal{L}_G\mathbf{x}}{\mathbf{x}^T\mathbf{x}}|\mathbf{x}\perp\mathbf{u}, \mathbf{x}\neq 0)$$

. Since H is always denser than G, according to Courant-Weyl Inequalities, $\alpha \geq 1$[40]. ■

For convenience, we denote the

$$\gamma = \Omega(p_g p_s(1+\alpha)\frac{a(G)}{d}).$$

Since groups are randomly formed in the network, every round may have various possible group distribution.

Lemma 4: Let the **conditional** expectation value of ϕ_k computed over all possible group distributions in round k given an group distribution with the potential ϕ_{k-1} in the previous round $k-1$ is $E_{\partial_k}[\phi_k]$. Here we denote the $\partial_1, \partial_2, \ldots, \partial_k$ as the independent random variables representing the possible group distributions happening at rounds 1, 2, ..., k, respectively. Then, the $E[\phi_k] = E_{\partial_1,\partial_2,\ldots,\partial_k}[\phi_k] \leq (1-\gamma)^k\phi_0$.

Proof: From the Theorem 3, the

$$E_{\partial_k}[\phi_k] \leq (1-\gamma)\phi_{k-1}$$

and by the definition,

$$\begin{aligned} E[\phi_k] &= E_{\partial_1,\partial_2,\ldots,\partial_k}[\phi_k] \\ &= E_{\partial_1}[E_{\partial_2}[\cdots E_{\partial_{k-1}}[E_{\partial_k}[\phi_k]]\cdots]] \\ &\leq (1-\gamma)E_{\partial_1}[E_{\partial_2}[\cdots E_{\partial_{k-1}}[\phi_{k-1}]\cdots]] \\ &\vdots \quad \vdots \\ &\leq (1-\gamma)^k\phi_0 \end{aligned}$$

■

The next proposition relates the potential to the accuracy criterion.

Proposition 5: Let ϕ_τ be the potential right after the τ-th round of the DRG Ave algorithm, if $\phi_\tau \leq \varepsilon^2$, then the consensus has been reach at or before the τ-th round.

(the potential of the τ-th round $\phi_\tau \leq \varepsilon^2 \rightarrow |v_i^{(\tau)} - \bar{v}| \leq \varepsilon,\ \forall i$)

Proof: The v_i and $\bar{v}$ in the following are the value on node i and the average value over the network respectively, right after round τ.

$$\begin{aligned} \because \quad & (v_i - \bar{v})^2 \geq 0, \ \forall i \in \mathcal{V}(G) \\ \therefore \quad & \phi_\tau = \sum_{i \in \mathcal{V}(G)} (v_i - \bar{v})^2 \leq \varepsilon^2 \\ & \Rightarrow (v_i - \bar{v})^2 \leq \varepsilon^2 \\ & \Leftrightarrow |v_i - \bar{v}| \leq \varepsilon, \ \forall i \in \mathcal{V}(G)) \end{aligned}$$

■

The proof of Theorem 1: Now we finish the proof of our main theorem.

Proof: By Lemma 4 and Proposition 5,

$$E[\phi_\tau] \leq (1-\gamma)^\tau \phi_0 \leq \varepsilon^2.$$

Taking logarithm on the two right terms,

$$\begin{aligned} \tau \log(\frac{1}{1-\gamma}) &\geq \log \phi_0 - \log \varepsilon^2 \\ \tau &\geq \frac{\log(\frac{\phi_0}{\varepsilon^2})}{\log(\frac{1}{1-\gamma})} \\ &\approx \frac{1}{\gamma} \log(\frac{\phi_0}{\varepsilon^2}) \end{aligned} \tag{3}$$

W.l.o.g, we can assume $\phi_0 \gg \varepsilon^2$, otherwise the accuracy criterion is trivially satisfied. By Markov inequality

$$Pr(\phi_\tau > \varepsilon^2) < \frac{E[\phi_\tau]}{\varepsilon^2} \leq \frac{(1-\gamma)^\tau \phi_0}{\varepsilon^2}$$

Choose $\tau = \frac{\kappa}{\gamma} \log(\frac{\phi_0}{\varepsilon^2})$ where the $\kappa \geq 2$. Then because $(\frac{\varepsilon^2}{\phi_0}) \ll 1$ and $(\kappa - 1) \geq 1$,

$$\begin{aligned} Pr(\phi_\tau > \varepsilon^2) &< \frac{(1-\gamma)^{\frac{\kappa}{\gamma} \log(\frac{\phi_0}{\varepsilon^2})} \phi_0}{\varepsilon^2} \\ &\simeq e^{-\log(\frac{\phi_0}{\varepsilon^2})^\kappa} \frac{\phi_0}{\varepsilon^2} \\ &= (\frac{\varepsilon^2}{\phi_0})^{(\kappa-1)} \to 0 \, . \end{aligned}$$

Since with high probability $\phi_\tau \leq \varepsilon^2$ when $\tau = O(\frac{1}{\gamma} \log \frac{\phi_0}{\varepsilon^2})$, by proposition 5 the accuracy criterion must have been reached at or before the τ-th round. ■

B. Discussion of the upper bound in Theorem 1

As mentioned earlier, p_s is related to p_g and the topology of the underlying graph. For example, in a Poisson random geometric graph[50], in which the location of each sensor node can be modeled by a 2-D homogeneous Poisson point process with intensity λ, $p_s = e^{-\lambda \cdot p_g \cdot 4\pi r^2}$, where r is the transmission range. We can simply assume that sensor nodes are deployed in an *unit area*, so that λ is equal to n. To maintain the connectivity, we set

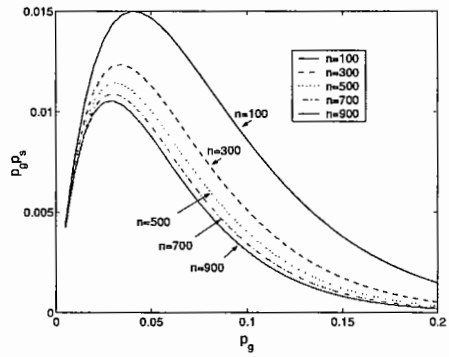


Fig. 3. The $\mathcal{P} = p_g p_s$ vs p_g on the Poisson random geometric graph

$4\pi r^2 = \frac{z(n)}{n} = 4\frac{\log(n)+\log(\log(n))}{n}$[36]. Let $\mathcal{P} = p_g p_s$. The maximum of $\mathcal{P} = p_g e^{-p_g \cdot z(n)}$, denoted as $\hat{\mathcal{P}}$, happens at $\hat{p}_g = \frac{1}{z(n)} = \frac{1}{4(\log(n)+\log(\log(n)))}$ where $\frac{d\mathcal{P}}{dp_g} = 0$. The maximum $\hat{\mathcal{P}} \simeq \frac{1}{4d}e^{-1}$.

Fig.3 shows the curves of $p_g p_s$ on Poisson random geometric graphs with different n from 100 to 900. It is easy to find the $\hat{p}_g$ in these graphs. For instance, given a Poisson random geometric graph with $n = 500$, we can choose the $\hat{p}_g \simeq 0.03$ so that DRG will expectedly converge fastest, for a given set of other parameters.

In general, for an arbitrary graph $\mathcal{P} = p_g(1-p_g)^{\chi}$; where $\chi = O(d^2)$ is the expected number of nodes within two hops of the group leader. Then the $\hat{\mathcal{P}} \simeq \chi^{-1}e^{-1}$, happens when $\hat{p}_g = \chi^{-1}$. For instance, a *d-regular* expander, the $\hat{p}_g = \frac{1}{d^2}$ and $\hat{\mathcal{P}} \simeq \frac{1}{d^2}e^{-1}$.

Fixing the $p_g = \frac{1}{d^2}$, we get $\mathcal{P} = \frac{1}{d^2}e^{-\frac{O(d^2)}{d^2}} < \frac{1}{d^2}$. Hence, we get a ***ultimate upper bound of the expected running time of DRG for any connected graph***:

$$O(\frac{d^3 \log(\frac{\phi_0}{\varepsilon^2})}{(1+\alpha)a(G)}).$$

If we specify a graph and know its χ, by carefully choosing p_g to maximize $\mathcal{P} = p_g p_s$, we can get a tighter bound for the graph than the bound above. Thus, we want to emphasize the bound in Theorem 1 because it can be much tighter according to the given topology and reflects the topology's characteristic.

C. The upper bound of the expected number of total transmissions

Lemma 6: For DRG algorithm, the expected total number of transmissions in a round is $E[T_r] = O(p_g p_s dn)$, where $d = \max(d_i) + 1$.

Proof: Since the necessary transmissions for a group g_i to locally compute aggregate is $d_i + 2$, bounded by $d+1 \approx d$, the expected total number of transmissions in a round is $O(p_g p_s dn)$. ■

Theorem 7: Given a connected undirected graph G with the initial potential ϕ_0, the total expected number of necessary transmissions for the value distribution to reach the consensus in an accuracy requirement ε^2 is

$$E[T] = O(\frac{nd^2 \log(\frac{\phi_0}{\varepsilon^2})}{(1+\alpha)a(G)})$$

Proof:

$$\begin{aligned} E[T] &= E[T_r]O(\frac{d\log(\frac{\phi_0}{\varepsilon^2})}{p_g p_s(1+\alpha)a(G)}) \\ &= O(p_g p_s\, n\,(d+1)\frac{d\log(\frac{\phi_0}{\varepsilon^2})}{p_g p_s(1+\alpha)a(G)}) \\ &= O(\frac{nd^2\log(\frac{\phi_0}{\varepsilon^2})}{(1+\alpha)a(G)}) \end{aligned}$$

■

D. DRG Max/Min Algorithms

Instead of announcing the local average of a group, the group leader in the DRG Max/Min algorithm announces the local Max/Min of a group. Then all the members of a group update their values to the local Max/Min. Since the global Max/Min is also the local Max/Min, the global Max/Min value will progressively replace all the other values in the network.

In this subsection, we analyze the running time of DRG Max/Min algorithms by using the analytical results of the DRG Ave algorithm. However, for the Max/Min we need a different accuracy criterion: $\rho = \frac{n-m}{n}$, where n,m is the total number of nodes and the number of nodes of the global Max/Min, respectively. ρ indicates the proportion of nodes that have *not yet* changed to the global Max/Min. When a small enough ρ is satisfied after running DRG Max/Min, with high probability $(1-\rho)$, a randomly chosen node is of the global Max/Min.

We only need to consider Max problem since Min problem is symmetric to the Max problem. Moreover, we assume there is only one global Max value v_{max} in the network. This is the worst situation. If there is more than one node with the same v_{max} in the network then the network will reach consensus faster because there is more than one "diffusion" source.

***Theorem** 8:* Given a connected undirected graph $G(\mathcal{V},\mathcal{E})$, $|\mathcal{V}| = n$ and an arbitrary initial value distribution $\mathbf{v}^{(0)}$, then with high probability the Max/Min problem can be solved under the desired accuracy criterion ρ, after invoking the DRG Max/Min Algorithm

$$O(\frac{1}{\gamma}\log(\frac{(1-\rho)n}{\rho}))$$

times, where the $\gamma = \Omega((1+\alpha)a(G)\frac{p_g p_s}{d})$.

Proof: The proof is based on two facts: (1) The expected running time of the DRG Max/Min algorithm on an arbitrary initial value distribution $\mathbf{v}_a^{(0)} = (v_1, \ldots, v_{i-1}, v_i = v_{max}, v_{i+1} \ldots, v_n)^T$ will be exactly the same as that on the binary initial distribution $\mathbf{v}_b^{(0)} = (0,\ldots,0, v_i = 1, 0,\ldots 0)^T$ under the same accuracy criterion ρ. The v_{max} in $\mathbf{v}_a^{(0)}$ will progressively replace all the other values no matter what the replaced values are. We can map the v_{max} to "1" and all the others to "0". Therefore, we only need to consider the special binary initial distribution $\mathbf{v}_b^{(0)}$ in the following analysis. (2) Suppose the DRG Ave and DRG Max algorithms are running on the same binary initial distribution $\mathbf{v}_b^{(0)}$ and going through the same grouping scenario which means that the two algorithms encounter the same group distribution in every round. Under the same grouping scenario, in each round, those nodes of value "0" in DRG Ave are also of value "0" in DRG Max.

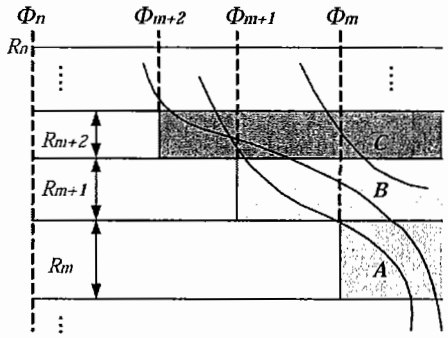


Fig. 4. The minimum potential

Based on these two facts, a relationship between two algorithms' accuracy criteria: $\varepsilon^2 = \frac{\rho}{(1-\rho)n}$, can be exploited to obtain the upper bound of expected running time of DRG Max algorithm from that of DRG Ave algorithm. Now we present our analysis in detail.

We run two algorithms on the same initial value distribution $\mathbf{v}_b^{(0)}$ and go through the same scenario. To distinguish their value distributions after, say ζ rounds, we denote the value distribution for DRG Ave as $\mathbf{v}^{(\zeta)} \equiv \mathbf{v}_b^{(\zeta)}|_{DRG\ Ave}$ and that for DRG Max as $\mathbf{w}^{(\zeta)} \equiv \mathbf{v}_b^{(\zeta)}|_{DRG\ Max}$.

Without loss of generality, suppose $\mathbf{w}^{(\zeta)} = (w_1 = 1, \ldots, w_m = 1, w_{m+1} = 0, \ldots, w_n = 0)^T$. There are m "1"s and $(n-m)$ "0"s. Then the corresponding $\mathbf{v}^{(\zeta)} = (v_1, v_2, \ldots v_m, v_{m+1} = 0, \ldots, v_n = 0)^T$. Apparently $w_i = \lceil v_i \rceil$. Although the values from v_{m+1} to v_n are still "0"s, the values from v_1 to v_m could be any value $\in (0,1)$. To bound the running time, we need to know the potential ϕ_ζ, which now is a random variable at the ζ-th round. We now calculate a bound on the minimum value for the potential ϕ_ζ.

The minimum value of the potential ϕ_ζ at the ζ- round with exactly m non-zero values is a simple optimization problem formulated as follows:

$$\begin{aligned} \textbf{min} \quad & \sum_{i \in \mathcal{V}(G)} (v_i - \overline{v})^2 \\ \text{subject to} \quad & \sum_{i=1}^{m} v_i - 1 = 0 \\ & 1 \geq v_i \geq 0; \quad 1 \leq i \leq m, \\ & v_i = 0; \quad m < i \leq n. \end{aligned}$$

where $n = |\mathcal{V}(G)|$ and $\overline{v} = \frac{1}{n}$.

By the Lagrange Multiplier Theorem, the minimum happens at

$$v_i^* = \begin{cases} \frac{1}{m} & 1 \leq i \leq m. \\ 0 & \text{otherwise.} \end{cases}$$

and the ***minimum potential*** is $\phi_\zeta^* = \frac{1}{m} - \frac{1}{n}$.

Each round ζ is associated with a value distribution $\mathbf{v}^{(\zeta)}$. We define a set R_m as the set of rounds which are of m non-zero values in their value distributions. $R_m = \{\zeta | \mathbf{v}^{(\zeta)}$ is of m non-zero value$\}$ and the ***minimum potential***

$$\Phi_m = \min(\phi_\zeta) = \frac{1}{m} - \frac{1}{n}, \forall \zeta \in R_m$$

The possible scenarios A, B and C are shown in Fig.4. The y-axis is the time episode in the unit of a round, we group those rounds by R_m as defined earlier. The x-axis is the potential of each round. Note that the value of each round are not continuous. The scenario curves A, B, and C just show the decreasing trend of potentials. The scenario A reaches the minimum potential of R_m at its last round in R_m. For scenario A, the diffusion process is slower, while the value distribution is more balanced over nodes.

Proposition *9:* A round ζ of DRG Ave algorithm with distribution $\mathbf{v}^{(\zeta)}$ and potential ϕ_ζ, if $\phi_\zeta \leq \Phi_m$ then there are **at least** m non-zero value within $\mathbf{v}^{(\zeta)}$.

$(\phi_\zeta \leq \Phi_m \rightarrow |S| \geq m,\ S = \{v_i | v_i^{(\zeta)} > 0\})$

Proof: A round ζ is with $\phi_\zeta \leq \Phi_m$ but has less than m non-zero value tuples in $\mathbf{v}^{(\zeta)}$. WLGN, suppose there are $m-1$ nonzero values in $\mathbf{v}^{(\zeta)}$, then $\phi_\zeta \geq \Phi_{m-1}$. But $\Phi_m < \Phi_{m-1}$. A contradiction. ■

By the fact that there are m non-zero values in $\mathbf{v}^{(\zeta)}$ if and only if there are m "1"s in $\mathbf{w}^{(\zeta)}$ and by proposition 9, we can set

$$\Phi_m = \varepsilon^2 = \frac{1}{m} - \frac{1}{n} = \frac{\rho}{(1-\rho)n}.$$

Substituting $\frac{\rho}{(1-\rho)n}$ for ε^2 in Theorem 1, we get the upper bound of the expected running time of DRG Max algorithm to reach a desired accuracy criterion $\rho = \frac{n-m}{n}$, which is

$$O(\frac{1}{\gamma}\log(\frac{(1-\rho)n}{\rho})).$$

The γ follows the rules mentioned before.

The upper bound of the expected number of the total necessary transmissions for DRG Ave is

$$E[T] = O(\frac{nd^2\log(\frac{(1-\rho)n}{\rho})}{(1+\alpha)a(G)})$$

by the same deriving process of Theorem 7.

E. Bounds for Poisson random geometric graph

The analysis results presented are for arbitrary graphs. Also, the assumption of complete groups in analysis is a worst situation setting for deriving an ultimate upper bound. If we specify a graph and know the distribution of nodes, besides complete groups, we can further count the partial groups to get a *tighter* upper bound for the expected running time.

In a variety of applications on sensor network, sensor nodes are randomly scattered in a field. The sensor nodes are then uniformly distributed in the field, which means that the location of a sensor node follows a 2D Homogeneous Poisson Point Process. The underlying graph of this kind of sensor network is called Poisson random geometric

graph. Since we know the nodes' distribution of a Poisson random geometric graph, we develop a tighter bound from relaxing the complete group assumption.

For a Poisson random geometric graph with intensity λ, every node independently announces to be a group leader in a probability p_g, so that the distribution of leader nodes is also a 2D Homogeneous Poisson point process with intensity $\lambda_l = \lambda \cdot p_g$. Further, to keep the connectivity of the underlying graph, the radius of the radio coverage, r, is a function of λ and can be denoted as $r(\lambda)$. The probability p_s, is the probability that a GCM from a leader node i encounters no collisions. This means that no nodes other than i within an area $\pi(2r(\lambda))^2$ centered at i become group leaders. Hence, the probability $p_s = e^{-\lambda_l \cdot 4\pi r(\lambda)^2} = e^{-p_g \cdot \lambda \cdot 4\pi r(\lambda)^2}$ is a function of the p_g and the intensity λ.

As in Fig.5, if two nodes announce to be leaders simultaneously, a GCM collision happens in the overlap area of their radio coverage. Both their GCMs are destroyed due to the radio interference. All nodes within the overlap area can not correctly receive their GCMs. However, those nodes not in the overlap area can still correctly receive and response to a GCM from one of the two leader nodes.

In the following we relax the consideration in the previous subsection that only those complete groups are counted in the computation of $\delta\phi$. A group g_i may only contain part of the neighboring nodes of node i. We name this kind of groups as *partial groups*. Since nodes are distributed by uniform distribution over the field, the number of members in a group led by leader node i is $|g_i| = \frac{\mathcal{A}_i}{\pi r^2} \cdot N_G(i) + 1$, where $\mathcal{A}_i$, as shown in Fig 5, is the area in which GCM from leader node i can be successfully received. When node i becomes a group leader, in average there will be $p_g \lambda \pi (2r)^2$ other group leaders announcing their GCMs simultaneously causing GCM collisions with node i. Since the graph is connected, $\pi r^2 = \frac{\log \lambda + c(\lambda)}{\lambda}$ [36]. Let $p_g = O(\frac{1}{4(\log \lambda + c(\lambda))})$, then in average there is only one group leader other than i to cause GCM collision with node i. Below, we derive a tighter upper bound for the running time of DRG Ave algorithm. This upper bound of the running time is of the same form as *Theorem*1 but p_s needs to be modified as $\tilde{p}_s$:

$$\tilde{p}_s = p_s + \int_0^{2r} f_{\mathbf{y}}(y) Q_i(y)^2 dy,$$

where $Q_i(y)$ is the ratio of non-overlapping area ($\mathcal{A}_i(y)$ in Figure 5(c)) [3] to the radio coverage πr^2. The upper bound in ***Theorem*** 1 becomes

$$O(\frac{\log(\frac{\phi_0}{\epsilon^2})}{p_g \tilde{p}_s (1+\alpha) a(G)}).$$

We detail the deriving process in the following paragraphs.

Let the random variable $\mathbf{y}$ be the distance between a group leader i, the origin, and the first another group leader, both announcing group call messages at the same time. When $\mathbf{y} \leq 2r$, see Fig.5 (a)(b), the area of the group

[3]It may be too pessimistic to assume that the collision will totally destroy the group call messages. If FM is used, nodes always can extract the stronger signal, known as the Capture Effect[51]. The group coverage then becomes Fig. 5(d). However, since there is not standards established for the PHY and MAC layers in sensor networks yet, we would like to analyze our algorithm in a more general way. We assume the shaded area in Fig. 5(c) to be the coverage of a group facing group call massage collision.

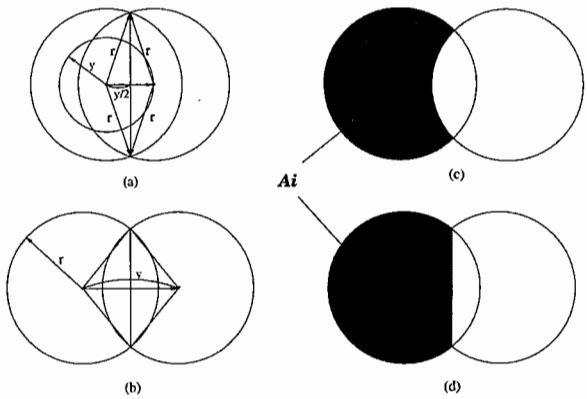


Fig. 5. The overlapping of two groups

coverage is $\mathcal{A}_i(\mathbf{y})$, the shaded area in Fig. 5(c). Let $Q_i(\mathbf{y})$ be the ratio of group coverage $\mathcal{A}_i$ to the radio coverage πr^2: $Q_i(\mathbf{y}) = \frac{\mathcal{A}_i(\mathbf{y})}{\pi r^2}$, $0 \leq Q_i(\mathbf{y}) \leq 1$, where $0 \leq \mathbf{y} \leq 2r$.

The PDF of $\mathbf{y}$ is $f_{\mathbf{y}}(y) = 2\pi\lambda_l y e^{-\pi\lambda_l y^2}$.

The expected decrement of potential in the group g_i, is

$$E[\delta\varphi_i] = \int_0^{2r} \delta\varphi_i(y) \cdot f_{\mathbf{y}}(y)dy + \int_{2r}^{\infty} \delta\varphi_i(y) \cdot f_{\mathbf{y}}(y)dy. \tag{4}$$

The first item in the R.H.S. involves GCM collisions while the second does not. Obviously,

$$\int_{2r}^{\infty} \delta\varphi_i(y) \cdot f_{\mathbf{y}}(y)dy = \frac{p_g p_s}{d_i + 1} \sum_{(j,k)\in\mathcal{E}(C_i)} \Delta_{jk},$$

where $p_s = e^{-\lambda_l 4\pi r^2}$.

If node i becomes a group leader, the conditional probability that a neighboring node j joins its group is $Q_i(y)$. Thus the Δ_{ij} will contribute to the potential decrement $\delta\varphi_i$ in a conditional probability $Q_i(y)$. Also, if two neighboring nodes j and k both join i's group, then the Δ_{jk} will contribute to the the potential decrement $\delta\varphi_i$, no matter (j,k) is the true edge or the auxiliary edge. The conditional probability that both j and k join i's group is $Q_i(y)^2$.

Let $\Gamma = \sum_{(i,j)\in\mathcal{E}(C_i)} \Delta_{ij}$ and $\Upsilon = \sum_{\substack{(j,k)\in\mathcal{E}(C_i) \\ j,k\neq i}} \Delta_{jk}$. Then

$$\begin{aligned} \sum_{(j,k)\in\mathcal{E}(C_i)} \Delta_{jk} &= \sum_{(i,j)\in\mathcal{E}(C_i)} \Delta_{ij} + \sum_{\substack{(j,k)\in\mathcal{E}(C_i) \\ j,k\neq i}} \Delta_{jk} \\ &= \Gamma + \Upsilon. \end{aligned}$$

We thus obtain the first term of R.H.S. of equation (4) as follows.

$$\begin{aligned}\int_0^{2r} \delta\varphi_i(y) \cdot f_{\mathbf{y}}(y)dy &= p_g \int_0^{2r} f_{\mathbf{y}}(y)(\frac{Q_i(y)\Gamma + Q_i(y)^2\Upsilon}{Q_i(y)d_i + 1})dy \\ &\geq p_g \int_0^{2r} f_{\mathbf{y}}(y)(\frac{Q_i(y)\Gamma + Q_i(y)^2\Upsilon}{d_i + 1})dy \\ &\geq \frac{p_g}{d_i+1}\int_0^{2r} f_{\mathbf{y}}(y)(Q_i(y)^2\Gamma + Q_i(y)^2\Upsilon)dy \\ &= \frac{p_g \sum_{(j,k)\in\mathcal{E}(C_i)} \Delta_{jk}}{d_i+1}\int_0^{2r} f_{\mathbf{y}}(y)Q_i(y)^2dy.\end{aligned}$$

Therefore, the

$$E[\delta\varphi_i] \geq \frac{p_g \sum_{(j,k)\in\mathcal{E}(C_i)} \Delta_{jk}}{d_i+1}(p_s + \int_0^{2r} f_{\mathbf{y}}(y)Q_i(y)^2dy)$$

Let

$$\tilde{p}_s = p_s + \int_0^{2r} f_{\mathbf{y}}(y)Q_i(y)^2dy$$

and $d = \max(d_i) + 1$, so that

$$E[\delta\varphi_i] \geq \frac{p_g\tilde{p}_s \sum_{(j,k)\in\mathcal{E}(C_i)} \Delta_{jk}}{d}$$

and

$$\begin{aligned}E[\delta\phi] &= \sum_{i\in\mathcal{V}(G)} E[\delta\varphi_i] \\ &\geq p_g\tilde{p}_s\frac{1}{d}\sum_{i\in\mathcal{V}}\sum_{(j,k)\in\mathcal{E}(C_i)} \Delta_{jk}\end{aligned}$$

which is exactly of the same form of the equation (1) but with a modification of replacing p_s with $\tilde{p}_s$. The same analyzing procedures in the previous subsection can be repeated. The coverage ratio becomes

$$\tilde{\gamma} = \Omega((1+\alpha)a(G)\frac{p_g\tilde{p}_s}{d})$$

and the upper bound of expected running time is

$$O(\frac{1}{\tilde{\gamma}}\log(\frac{\phi_0}{\eta})).$$

Since $\tilde{p}_s \geq p_s$, this bound is tighter than the general bound in Theorem 1.

F. Random grouping with link failures

Wireless links amid sensor nodes, which are *edges* in the underlying graph, may fail due to natural or adversary interferences and obstacles. With a modification on the potential convergence ratio we obtain modified upper bounds for the expected performance of DRG while involving possible link failures.

We assume that the failure of a wireless link happens only between grouping time slots. Let $\acute{G}$ be a subgraph of G, obtained by removing the failed edges from G at the end of the algorithm and $\acute{H}$ be the auxiliary graph of $\acute{G}$. Lemma 3 can be modified as the following:

***Lemma** 10:* Given a connected undirected graph G, The potential convergence ratio involving edge failures is

$$E[\frac{\delta\phi}{\phi}] \geq \frac{p_g p_s}{d}(1+\acute{\alpha})a(\acute{G})$$

where the $\acute{G}$ is a subgraph of G, obtained by removing the failed edges from G at the end of the algorithm, and $\acute{\alpha} = \frac{a(\acute{H})}{a(\acute{G})}$.

Proof: Let $G^{(k)}$ be the underlying graph of sensor network after running DRG for k rounds but not yet reaching the convergence. $G^{(k)}$ is a subgraph of G excluding those failed edges from G. By the facts:

1) the maximum degree $d = d(G) \geq d(G^{(k)}) \geq d(\acute{G})$,
2) $a(G) \geq a(G^{(k)}) \geq a(\acute{G})$ and $a(H) \geq a(H^{(k)}) \geq a(\acute{H})$,

we have

$$E[\frac{\delta\phi_k}{\phi_k}] \geq \frac{p_g p_s}{d(G^{(k)})}(a(G^{(k)}) + a(H^{(k)})) \geq \frac{p_g p_s}{d}(a(\acute{G}) + a(\acute{H})) = \frac{p_g p_s}{d}(1+\acute{\alpha})a(\acute{G}).$$

■

By Lemma 10, we obtain the modified convergence ratio $\acute{\gamma} = \frac{p_g p_s}{d}(1+\acute{\alpha})a(\acute{G})$. Replacing γ by $\acute{\gamma}$ we have the modified upper bounds of performance of DRG in case of edge failures.

VI. Extensions and Practical Considerations

We can easily compute some other aggregates like Count and Sum using DRG Ave as described in the following subsection. Moreover, if nodes can pre-process their values then we can compute much more complex functions over the network. For example, to compute the function $f(\mathbf{v}) = \sum c_i f_i(v_i)$, each node i can first process its original value v_i to a desired local function $c_i f_i(v_i)$. Then by applying our DRG Sum algorithm, we can get the grand sum $\sum c_i f_i(v_i)$ over the network.

A. Algorithms for Other Aggregates

1) DRG Count: We can solve the count problem using the DRG Ave algorithm. First, the network should run a health check in the background by DRG Ave periodically to know the current number of alive nodes, n. The sink node sets its initial value to "1" and all the other nodes set their initial value as "0". After applying the DRG Ave algorithm, all the nodes' values will converge to a value $\frac{1}{n}$. Thus we get n. When n is less than some threshold, we will need to replenish the sensor nodes by adding more sensor nodes to the field.

Secondly, to count the number of nodes with a certain property (e.g. battery voltage lower than a threshold or temperature higher than an alert threshold), all nodes of the property will set their initial values as "1", while the others set their initial values as "0". Running the DRG Ave, progressively, all the nodes will reach the average value $\frac{m}{n}$, where the m is the number of nodes of the inquired property. Any node can know the $count = m$ by multiplying this converged value $Ave = \frac{m}{n}$ by the number of alive nodes n, which is known from regular network health checking.

2) DRG Sum: The Sum problem can also be solved by applying DRG Ave. When every node knows the average value Ave, they can just multiply it by the number of alive nodes n to get the $Sum = n \cdot Ave$. If we want to sum the values of some nodes of a certain property, first we count the number of these nodes by the DRG Count algorithm and then apply the DRG Ave. However, during the DRG Ave, the nodes which are not of the property set their initial value as "0". Then multiplying the convergent average value Ave by the $Count\ m$, all the nodes can know the sum of the values of some certain property.

B. Practical Considerations

A practical issue is deciding when nodes should stop the DRG iterations of a particular aggregate computation. An easy way to stop, as in [7], is to let the node which initiates the aggregate query disseminate a stop message to cease the computation. The querying node samples and compares the values from different nodes located at different locations. If the sampled values are all the same or within some satisfiable accuracy range, the querying node disseminates the stop messages. This method incurs a delay overhead on the dissemination. To analyze the accuracy of final values and the probability of getting a false stop in this manner is left for future work.

A purely distributed local stop mechanism on each node is also desirable. The related distributed works [6], [7], [31] all fail to have such a local stop mechanism. However, nodes running our DRG algorithms could stop the computation locally. The purely local stop mechanism is to adapt the grouping probability p_g to the value change. If in consecutive rounds, the value of a node remains the same or just changes within a very small range, the node reduces its own grouping probability p_g accordingly. When all the nodes meet the accuracy criterion, they will just virtually stop their computation. However, being passive due to low p_g, a node will still join a group called by his neighbor. If the value changes again by an GAM, Group Assignment Message, from one of its neighbors, its grouping probability increases accordingly to actively re-join the aggregate computing process. We leave the detail of this implementation for future work.

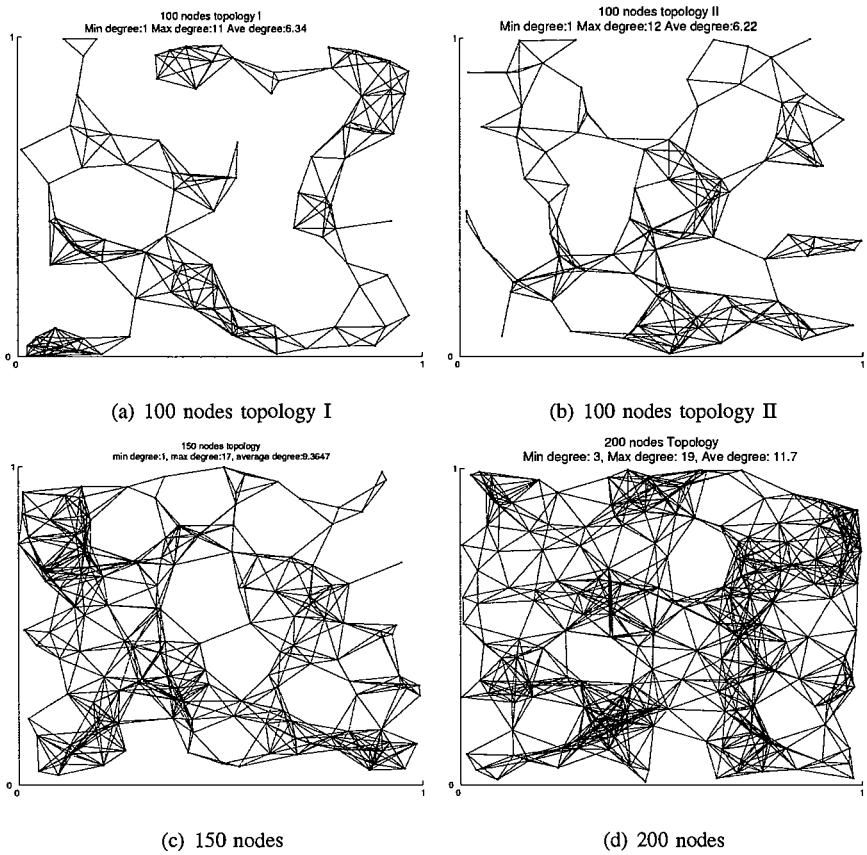
VII. Simulation Results

A. Experiment Setup

We performed simulations to investigate DRG's performance and numerically compare it with two other proposed distributed algorithms on grids and four instances of Poisson random geometric graph shown in Fig.6. Our simulations focus on the Average problem. We assume that the value v_i on each node follows an uniform distribution in an interval $\mathcal{I} = [0, 1]$. (DRG's performance on a case of $\mathcal{I} = [0, 1], \varepsilon = 0.01$ is the same as on a case of $\mathcal{I} = [0, 100], \varepsilon = 1$ and so on. Thus, we only need to consider an interval $\mathcal{I} = [0, 1]$.) On each graph, each algorithm is executed 50 times to obtain the average performance metrics. We run all simulation algorithms until all the nodes meet the *absolute* accuracy criterion $|v_i - \hat{v}| \leq \varepsilon$ in three cases: $\varepsilon = 0.01, 0.05, 0.1$

B. Performance of DRG

For grid, the topology is fixed and so the running time and the total number of transmissions grow as the grid size increases. Note that in Fig.7(a) and Fig.7(b), the axis of the grid size is not linear. Also, more stringent accuracy



(a) 100 nodes topology I

(b) 100 nodes topology II

(c) 150 nodes

(d) 200 nodes

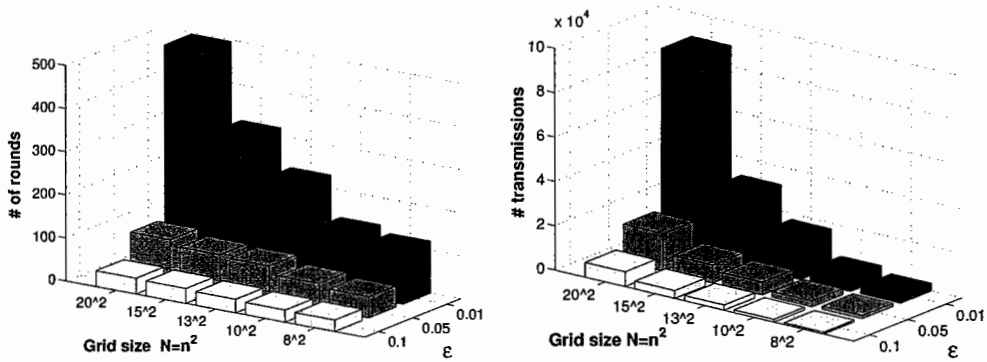
Fig. 6. The instances of Poisson random geometric graph used for simulations

requirement requires more running time and transmissions.

For Poisson random geometric graph, we observe that the topology significantly affects the performance. We have tried two different topologies each with 100 nodes. The 100 node topology I is less connected, implying that nodes in topology I have fewer options to spread their information. Thus, it is not surprising that both the total number of rounds and the total number of transmissions under topology I are much higher than those under topology II. In fact, the rounds and transmissions needed on 100-node topology I are even higher than on the instances of 150 nodes and 200 nodes in Fig.6. The two instances of 150 and 200 nodes are well connected and similar to the 100 nodes topology II. These results match our analysis where the parameters in the upper bound include not only the number of nodes n and grouping probability p_g, but also the parameters characterizing the topology — the maximum degree d and the algebraic connectivity $a(G)$.

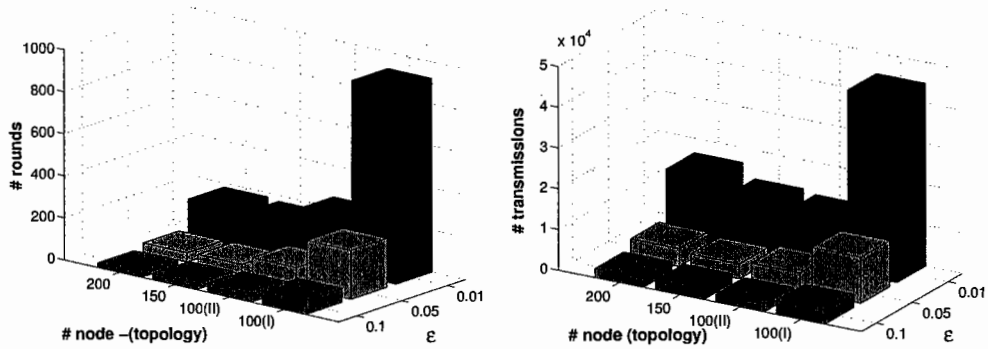
C. Comparison with Other Distributed Localized Algorithms

We briefly compare the performance of DRG with two other distributed localized algorithms for computing aggregates, namely, Flooding and Uniform Gossip[7]. In Flooding, each node divides its value and weight by d_i, its degree, and then broadcasts the quotient to all its neighbors (see Fig.10). In Uniform Gossip, each node randomly



(a) Running time on grid

(b) Total number of transmissions on grid



(c) Running time on Poisson random geometric graph

(d) Total number of transmissions on Poisson random geometric graph

Fig. 7. The Performance of DRG Ave on grid and Poisson random geometric graph.

picks one of its neighbors to send half of the value and weight and keeps the other half to itself. We numerically compare these two algorithms with DRG by simulations on grid and Poisson random geometric graphs.

We point out that the Flooding algorithm may never converge correctly to the desired aggregate on some topologies, e.g., a grid graph (since the graph is bipartite and hence the underlying Markov chain is not ergodic). Fig.8 is simple example to illustrate this pitfall. To solve this pitfall we propose a modified Flooding named Flooding-m in which each node i divides its value and weight by $d_i + 1$ and then send the quotient to "itself" and all its neighbors by broadcast. This modification incurs a more thorough and even mixing of values and weights on nodes, avoiding possible faulty convergence and expediting the running time.

Since different algorithms have their own definitions of "round", comparing running times by the number of rounds taken is not quite correct. In one round of Flooding-m or Uniform Gossip, there are n transmissions in which each node contributes one transmission. In a round of DRG, only those nodes in groups need to transmit data. The time duration of a round of DRG could be much shorter. Therefore, we compare DRG with Flooding-m and Uniform Gossip in terms of total number of transmissions. If three algorithms used the same underlying communication techniques (protocols), their expected energy and time costs for a transmission would be the same. Thus the total number of transmissions can be a measure of the actual running time and energy consumption.

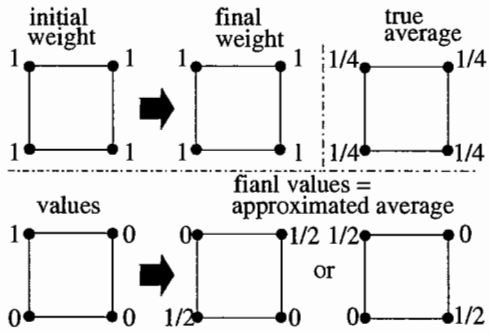
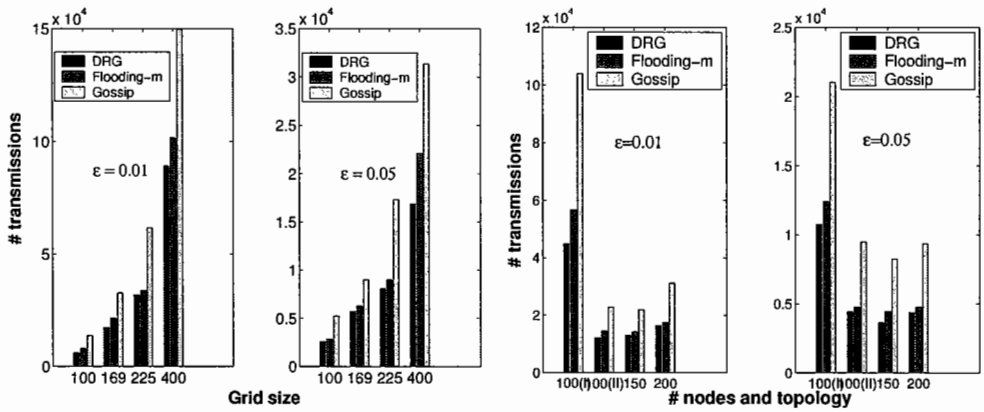


Fig. 8. An example that Flooding[7] can never converge to true average.

Uniform Gossip needs a much larger number of transmissions than DRG or Flooding-m. In grid, the topology is fixed, so the number of nodes is the only factor in the performance. The differences among the three algorithms increase while the grid size grows. On a grid of 400 nodes and $\varepsilon = 0.05$, DRG can save up to 25% of total number of transmissions than Flooding-m. In a random geometric graph, DRG can save up to 20% of total number of transmissions from Flooding-m on 100 nodes topology I under $\varepsilon = 0.01$. The trend is the same in the case when $\varepsilon = 0.1$.



(a) grid (b) Poisson random geometric graph

Fig. 9. The comparison of the total number of transmissions of 3 distributed algorithms

VIII. Conclusion

In this paper, we have presented distributed algorithms for computing aggregates through a novel technique of *random grouping*. Both the computation process and the computed results of our algorithms are naturally robust to possible node/link failures. The algorithms are simple and efficient because of their local and randomized nature, and thus can be potentially easy to implement on resource constrained sensor nodes.

We analytically show that the upper bound on the expected running times of our algorithms is related to the grouping probability, the accuracy criterion, and the underlying graph's spectral characteristics. Our simulation

Alg: Flooding

1 Initial: each node, e.g. node i sends $(s_{0,i} = v_i,\ w_{0,i} = 1)$ to itself.

2 Let $\{(\hat{s}_r,\ \hat{w}_r)\}$ be all pairs sent to i in round $t-1$.

3 Let $s_{t,i} = \sum_r \hat{s}_r;\ w_{t,i} = \sum_r \hat{w}_r$.

4 broadcast the pair $(\frac{s_{t,i}}{d_i},\ \frac{w_{t,i}}{d_i})$ to all neighboring nodes.

5 $\frac{s_{t,i}}{w_{t,i}}$ is the estimate of the average at node i of round t

Fig. 10. The broadcast-based Flooding algorithm [7]

results show that DRG Ave outperforms two representative distributed algorithms, Uniform Gossip and Flooding, in terms of total number of transmissions on both grid and Poisson random geometric graphs. The total number of transmission is a measure of energy consumption and actual running time. With fewer number of transmissions, DRG algorithms can be more resource efficient than Flooding and Uniform Gossip.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cyirci. "Wireless Sensor Networks: A Survey." Computer Networks, 38(4): 393-422, 2002.

[2] Chee-Yee Ching and Srikanta P. Kumar, "Sensor Networks: Evolution, Opportunities, and Challanges" Invited paper, Proceedings of The IEEE, Vol.91, No.8, Aug. 2003.

[3] J. Kulik, W. R. Heinzelman, H. Balakrishnan, "Negotiation-based protocols for disseminating information in wireless sensor networks", ACM/IEEE Int. Conf. on Mobile Computing and Networking, Seattle, WA, Aug. 1999.

[4] Deborah Estrin and Ramesh Govindan and John S. Heidemann and Satish Kumar, "Next Century Challenges: Scalable Coordination in Sensor Networks," International Conference on Mobile Computing and Networking, pp.263–270, 1999.

[5] J. M. Kahn and R. H. Katz and K. S. J. Pister, "Next Century Challenges: Mobile Networking for Smart Dust," International Conference on Mobile Computing and Networking, pp. 271–278, 1999.

[6] B. Ghosh and S. Muthukrishnan, "Dynamic load balancing by random matchings." J. Comput. System Sci., 53(3):357–370, 1996.

[7] D. Kempe A. Dobra J. Gehrke, "Gossip-based Computation of Aggregate Information", FOCS2003.

[8] Athanassios Boulis, Saurabh Ganeriwal, and Mani B. Srivastava, "Aggregation in Sensor Networks: An Energy- Accuracy Trade-off" Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications, 11 May 2003, pp. 128-138.

[9] Mihaela Enachescu, Ashish Goel, Ramesh Govindan, and Rajeev Motwani. "Scale Free Aggregation in Sensor Networks," The proceedings of the First International Workshop on Algorithmic Aspects of Wireless Sensor Networks (Algosensors), 2004.

[10] Nisheeth Shrivastava, Chiranjeeb Buragohain, Divyakant Agrawal, Subhash Suri. "Medians and Beyond: New Aggregation Techniques for Sensor Networks". ACM SenSys 2004.

[11] Suman Nath, Phillip B. Gibbons, Zachary Anderson, Srinivasan Seshan. "Synopsis Diffusion for Robust Aggregation in Sensor Networks" ACM SenSys 2004.

[12] Jerry Zhao, Ramesh Govindan, Deborah Estrin, "Computing Aggregates for Monitoring Wireless Sensor Networks," http://citeseer.nj.nec.com/zhao03computing.html

[13] Chaiporn Jaikaeo and Chavalit Srisathapornphat and Chien-Chung Shen, "Diagnosis of Sensor Networks", Helsinki, Finland, IEEE International Conference on Communications (ICC 2001), June, 2001.

[14] Jeremy Elson and Deborah Estrin, "Time Synchronization for Wireless Sensor Networks",Proceedings of the 2001 International Parallel and Distributed Processing Symposium (IPDPS), April 2001.

[15] Qun Li, Daniela Rus, "Global Clock Synchronization in Sensor Networks" Infocom, 2004, HongKong.

[16] Santashil PalChaudhuri, Amit Kumar Saha, David B. Johnson, "Adaptive Clock Synchronization in Sensor Networks" IPSN04, Berkeley, 2004.

[17] Philipp Blum,Lennart Meier,Lothar Thiele, "Improved Interval-Based Clock Synchronization in Sensor Networks" IPSN04, Berkeley, 2004.

[18] J. Heidemann, F. Silva, C. Intanagonwiwat, R. Govindan, D. Estrin, and D. Ganesan. "Building Efficient Wireless Sensor Networks with Low-Level Naming." In SOSP 2001.

[19] B. Krishnamachari, D. Estrin, and S. Wicker, "Impact of Data Aggregation in Wireless Sensor Networks," DEBS'02.

[20] S. Madden, M Franklin, J.Hellerstein W. Hong, "TAG: a tiny aggregation service for ad hoc sensor network," in proceeding of the USENIX OSDI Symposium 2002

[21] S.R.Madden, R. Szewczyk, M. J. Franklin, D Culler, "Supporting aggregate Queries over Ad-Hoc Wireless Sensor Networks," in Proc. of WMCSA 2002

[22] J. Gray , S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatrao, F. Pellow and H. Pirahesh "Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals," J. Data Mining and Knowledge Discovery, 29-53 (1997)

[23] Danny Raz and Yuval Shavitt, "New models and algorithms for programmable networks, Computer Net-works," Feb.2002 vol. 38(3) 311-326

[24] I.Gupta, R. Van Renesse, K.Birman, "Scalable fault-tolerant aggregation in large process groups," in Proc. Conf. On Dependable Systems and Networks, (2001)

[25] C. Intanagonwiwat, R. Govindan, and D. Estrin: Directed diffusion: A scalable and robust communication paradigm for sensor networks, Proceed.ing of the MobiCom (2000), 56-67

[26] R. van Renesse, K. P. Birman, W. Vogels: Astrolabe: A robust and scalable technology for distributed system monitoring, management, and data mining. ACM TOSC, 2003.

[27] M. BAWA, H. Garcia-Molina, A. Gionis, R. Motwani, "Estimating Aggregates on a Peer-to-Peer Network," Technical report, Computer Science Dept., Stanford University, 2003.

[28] H. Qi, S.S. Iyengar, and K. Chakrabarty, "Multi-resolution data integration using mobile agent in distributed sensor networks," IEEE Trans. Syst. Man, Cybern. C, vol. 31, pp. 383-391, Aug. 2001.

[29] Hairong Qi, Yingyue Xu, and Xiaoling Wang "Mobile-agent-based Collaborative Signal and Information Processing in Sensor Networks," PROCEEDINGS OF THE IEEE, VOL. 91, NO. 8, AUGUST 2003

[30] L. Tong, Q. Zhao, and S. Adireddy, "Sensor networks with mobile agents," in Proc. IEEE MILCOM'03, Boston, MA, October 2003.

[31] Dzulkifli Scherber,Babis Papadopoulos, "Locally Constructed Algorithms for Distributed Computations in Ad-Hoc Networks", IEEE IPSN04 Berkeley 2004.

[32] Joseph M. Hellerstein, Peter J. Haas, and Helen J. Wang, "Online Aggregation", In Proc. ACMSIGMOD International Conference on Management of Data, Tucson, May 1997.

[33] Weipeng P. Yan and Per-Ake Larson "Eager Aggregation and Lazy Aggregation," The VLDB Journal, pp.345-357,1995.

[34] E. Hung and F. Zhao, "Diagnostic Information Processing for Sensor-Rich Distributed Systems." Proc. of 2nd International Conference on Information Fusion (Fusion'99), Sunnyvale, CA, 1999.

[35] Josep Diaz, Jordi Petit, Maria J. Serna: A Random Graph Model for Optical Networks of Sensors. IEEE Transactions on Mobile Computing, pp.186-196 Volume 2, Number 3, July-September 2003

[36] P. Gupta and P. R. Kumar, "Critical power for asymptotic connectivity in wireless networks." Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of W.H. Fleming, W.M. McEneaney, G. Yin, and Q. Zhang (Eds.), Birkhauser, Boston, 1998.

[37] C. Bettstetter, "On the Minimum Node Degree and Conectivity of a Wireless Multihop Network", Proc. ACM MobiHoc 02, pp. 80-91, 2001.

[38] W. N. Anderson, Jr. and T.D. Morley, "Eigenvalues of the Laplacian of a graph. University of Maryland Technique Report TR-71-45, 1971.

[39] M. Fiedler. Algebraic connectivity of graphs. Czechoslovak Math. J., 23:298–305, 1973.

[40] Dragoš M. Cvetković, Michael Doob and Horst Sachs. Spectra of graphs, theory and application, Acedemic Press, 1980.

[41] Russell Merris. Laplacian Matrics of Graphs: A Survey, Linear Algebra Appl. 197/198 (1994) 143-176.

[42] R.Grone, Eigenvalues and the degree sequence of graphs, Linear and Multilinear Algebra 39 (1995), 133-136.

[43] B. Mohar, Some applications of Laplace eigenvalues of graphs, in "Graph Symmetry: Algebraic Methods and Applications," Eds. G. Hahn and G. Sabidussi, NATO ASI Ser. C 497, Kluwer, pp. 225-275.

[44] B. Mohar and S. Poljak, "Eigenvalues in Combinatorial Optimization," Res. Rep. 92752, IMA, Minneapolis, 1992.

[45] Stephen Guattery and Gary L. Miller, "Graph Embeddings and Laplacian Eigenvalues," SIAM Journal on Matrix Analysis and Applications vol.21, number 3,703–723, 2000.

[46] Stephen J. Kirkland, Michael Neumann, Bryan L. Shader. "Bounds on the subdominant eigenvalue involving group inverses with applications to graphs," Czechoslovak Mathematical Journal, Vol. 48, No. 1, pp. 1-20, 1998.

[47] Stephen J. Kirkland, Michael Neumann, Bryan L. Shader. "On a bound on algebraic connectivity: the case of equality" Czechoslovak Mathematical Journal, Vol. 48, No. 1, pp. 65-76, 1998.

[48] Christos Gkantsidis, Milena Mihail, Ellen W. Zegura: Spectral Analysis of Internet Topologies. Infocom 2003 San Franciso, CA, USA

[49] B.Bollobas, Random Graphs. Academic Press, 1985.

[50] Mathew Penrose, Random Geometric Graphs. Oxford University Press, 2003.

[51] Dimitri Bertsekas, Robert Gallager : Data Networks, Prentice-Hall, Inc.

[52] R. Motwani and P. Raghavan : Randomized Algorithms, Cambridge University Press 1995

[53] Pramod K. Varshney : Distributed Detection and Data fusion, Springer

[54] Fred Bauer, Anujan Varma, "Distributed algorithms for multicast path setup in data networks", IEEE/ACM Transactions on Networking, no. 2, Apr 1996 pp. 181-191

[55] L. Lovasz. "Random Walks on Graphs: A survey." Combinatorics Paul Erdos is Eighty Volume 2, p1-46.

[56] Ding Liu, Manoj Prabhakaran "On Randomized Broadcasting and Gossiping in Radio Networks", COCOON'2002, Singapore, August 2002, pp. 340-349.

[57] R. Olfati-Saber. "Flocking for Multi-Agent Dynamic Systems: Algorithms and Theory," Technical Report CIT-CDS 2004-005.

APPENDIX I
UN-CONVERGENCE OF FLOODING[7]

Authors of [7] provided and analyzed a deterministic Flooding algorithm in Fig.10 to compute aggregates in sensor networks. They map the graph of a network to a Markov chain and Flooding's broadcast transmissions to random walks on the Markov chain. When the steady state is reached, the total values and weights on a node j are $s_{t,j} = \pi_j \cdot \sum_{i\in\mathcal{V}} v_i$ and $w_{t,j} = \pi_j \cdot \sum_{i\in\mathcal{V}} 1 = \pi_j \cdot n$ respectively, where the π_j is the stationary state probability of each state (node). Thus, all nodes can have the global average $\bar{v} = \frac{s_{t,j}}{w_{t,j}} = \frac{\sum_{i\in\mathcal{V}} v_i}{n}$. However, the random walk can never reach steady state on a non-ergodic Markov chain, implying that by Flooding algorithm nodes can never reach the desired global average on the Markov chain's corresponding graph. For example, given a *bipartite* graph $G = (\mathcal{V}, \mathcal{E})$ in which $\mathcal{V}_1 \subset \mathcal{V}$, $\mathcal{V}_2 \subset \mathcal{V}$ and an edge $(v_i, v_j) \in \mathcal{E}$ iff $v_i \in \mathcal{V}_1$ and $v_j \in \mathcal{V}_2$, the Flooding will not converge if $\sum_{i\in\mathcal{V}_1} v_i \neq \sum_{j\in\mathcal{V}_2} v_j$. But each node in $\mathcal{V}_k$ will converge to $\frac{\sum_{i\in\mathcal{V}_k} v_i}{|\mathcal{V}_k|}$; $k = 1$ *or* 2. Fig.8 is a simple illustration of the un-convergence of Flooding algorithm.

APPENDIX II
OPTIMIZATION

Here we prove that the minimum potential is $\frac{1}{m} - \frac{1}{n}$ in the condition that there are exactly m nodes of non-zero values out of the n nodes after running the Slotted DRG Algorithm on a Binary Initial value problem for a while.

Since we here only consider those nodes of non-zero values. Thus we restrict the $\mathbf{v} \in \mathbb{R}^m$. The object function $f : \mathbb{R}^m \to \mathbb{R}$ and the constraint function $h : \mathbb{R}^m \to \mathbb{R}$. There is only one constraint function, so $\lambda \in \mathbb{R}^1$. The

Optimization Problem then be formulated as the following:

$$\begin{aligned} \min \quad & f(\mathbf{v}) = \sum_{i=1}^{m} v_i^2 - \frac{1}{n} \\ \text{subject to} \quad & h(\mathbf{v}) = \sum_{i=1}^{m} v_i - 1 = 0 \\ & 1 > v_i = \mathbf{v}(i) > 0; \quad for\ 1 \le i \le m \end{aligned}$$

The Lagrangian function

$$l(\mathbf{v}, \lambda) = \sum_{i=1}^{m} v_i^2 - \frac{1}{n} + \lambda(\sum_{i=1}^{m} v_i - 1)$$

For a local minimizer $\mathbf{v}^*$, there exist a λ^*, s.t.

$$\mathbf{D}l(\mathbf{v}^*, \lambda^*) = \mathbf{0}^T,$$

or

$$\nabla l(\mathbf{v}^*, \lambda^*) = \mathbf{0}$$

where $\mathbf{D}l(\mathbf{v}, \lambda)$ is the first order derivative of Lagrangian function. Thus we have

$$2v_i + \lambda^* = 0, \ \forall\, i$$

and

$$\sum_{i=1}^{m} v_i = 1.$$

Thus

$$-\frac{m\lambda^*}{2} = 1$$

$$v_i = \frac{1}{m}, \ \forall\, i$$

Therefore,$\mathbf{v}^* = \frac{1}{m}\mathbf{u} = (\frac{1}{m}, \frac{1}{m}, \cdots, \frac{1}{m})^T$, $\lambda^* = \frac{-2}{m}$ and the Minimum $f(\mathbf{v}^*) = \frac{1}{m} - \frac{1}{n}$, which is the minimum potential Φ_m of our analysis in Max/Min algorithm.

Further we check the Second-Order condition:

$$L(\mathbf{v}, \lambda) = F(\mathbf{v}) + \lambda H(\mathbf{v})$$

where $L(\mathbf{v}, \lambda)$, $F(\mathbf{v})$, and $H(\mathbf{v})$ are the Hessian Matrices of $l(\mathbf{v}, \lambda)$, $f(\mathbf{v})$, and $h(\mathbf{v})$ respectively.

Both $f(\mathbf{v})$ and $h(\mathbf{v})$ are twice continuously differentiable, i.e., $f(\mathbf{v}) \in \mathscr{C}^2$ and $h(\mathbf{v}) \in \mathscr{C}^2$.

$F(\mathbf{v}) = 2I_{mxm}$ and $H(\mathbf{v}) = \mathbf{0}_{mxm}$. Then $L(\mathbf{v}^*, \lambda^*) = 2I_{mxm}$ is positive definite.

$$\mathbf{v}^T L(\mathbf{v}^*, \lambda^*)\mathbf{v} = 2\sum_{1}^{m} v_i^2 > 0$$

for all $\mathbf{v} \neq 0$, which satisfies the Second-Order Sufficient Conditions. Thus the $\mathbf{v}^* = \frac{1}{m}\mathbf{u}$ is the strict local minimizer. Actually it is the global minimizer within the definition range of $\mathbf{v} : 0 < v_i < 1$ for all $1 \le i \le m$.